UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/301,885	04/29/1999	DARRELL R. COMMANDER	1857-00200	6193

22879 7590 12/15/2003

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
ALI, SYED J	
ART UNIT	PAPER NUMBER
2127	10

DATE MAILED: 12/15/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

4

Office Action Summary	**Application No.** 09/301,885	**Applicant(s)** COMMANDER, DARRELL R.
	Examiner Syed J Ali	**Art Unit** 2127

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *September 23, 2003*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-24 and 27-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-24 and 27-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

13)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

a) ☐ The translation of the foreign language provisional application has been received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .

4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: .

DETAILED ACTION

1. This office action is in response to Amendment B, paper number 9, which was received September 23, 2003. Claims 1-24 and 27-30 are presented for examination.

2. The text of those sections of Title 35, U.S. code not included in this office action can be found in a prior office action.

Claim Rejections - 35 USC § 103

3. Claims 1-3, 5 are rejected under 35 U.S.C. 103(a) as being unpatentable over Coyle et al. (USPN 6,314,114) (hereinafter Coyle) in view of Broadhurst (USPN 6,560,634) and further in view of Sharma et al. (USPN 6,182,109) (hereinafter Sharma).

As per claim 1, Coyle discloses a parallel processing network in which one or more processes can be spawned, comprising:

a plurality of computers coupled together by a communications link (col. 6 lines 52-59, "Distributed system 300 includes three nodes, NODE1, NODE2, and NODE3 that are communicatively coupled so that they can exchange data and information with each other"); and

process spawning logic included in one of said plurality of computers that automatically spawns processes (col. 3 lines 51-67, "As additional processing resources are required, process pool managers 204, 208, and 212 can spawn additional server processes. Alternatively, as fewer processing resources are required, process pool managers 204, 208 and 212 can reduce the number of active server processes").

Broadhurst discloses the following limitations not shown by Coyle, specifically process spawning logic that spawns processes in response to user specified criteria (col. 5 lines 43-64, "Upon the user at client computer 106 entering a domain name to search, client computer 106 sends the user input to query server 104 [sep 410]", "search engine 226 calculates the number of search sub-processes to spawn based on the number of domains to query submitted by client computer 106") and that determines whether sufficient resources are available to spawn the processes and, if not, spawns a reduced number of processes based on the number of available resources (col. 5 lines 43-64, "depending on the load of query server 104, search engine 226 may change the number of search sub-processes to spawn").

It would have been obvious to one of ordinary skill in the art to combine Coyle with Broadhurst since Coyle reduces the number of processes spawned when it would save system resources, not necessarily when the system cannot support additional processes. Broadhurst makes up for this deficiency by allowing the number of processes spawned to be reduced in response to a condition where there are not enough processing resources available to service a process, i.e. when the query server is suffering from a high load.

It is noted that Broadhurst discloses reducing the number of spawned processes in response to an overload condition on a single server, and not necessarily due to a lack of available processors. Sharma discloses determining whether an execution unit is available for servicing a process, and if not, rejecting the request (col. 2 lines 38-62, wherein a processor can be interpreted to be one type of execution unit).

It would have been obvious to one of ordinary skill in the art to add Sharma to the modified Coyle since Sharma further discloses a determination of whether there are enough processors available to service a request. The combination thereof improves upon the modified Coyle since Sharma would simply reject the request rather than attempt to reallocate resources to achieve a higher level of efficiency. To that end, the combination of Coyle, Broadhurst, and Sharma allow the reduction of the number of processes spawned in response to a lack of available resources, e.g. not enough execution units or processors, available to service the spawned processes.

As per claim 2, "Official Notice" is taken that a parallel processing network wherein a communications link may include a switch is well known and expected in the art. Specifically, there are several devices that are commonly used as communications links in a parallel processing network such as hubs, switches, routers, and bridges. Depending on the specific needs of the network in question, it would have been obvious to one of ordinary skill in the art to use a switch as one form of communication since it provides a way of directing data across a network while also maintaining a degree of security by not allowing other nodes of a network to listen to communication between other nodes. Other advantages and disadvantages of switches exist, and are well known in the art.

As per claim 3, Broadhurst discloses the parallel processing network of claim 1 wherein the user specified criteria includes a number of processes the spawning logic should spawn (col. 5 lines 43-64, "search engine 226 calculates the number of search sub-processes to spawn based on the number of domains to query submitted by client computer 106", wherein the query submitted by the client computer is created in response to user input).

As per claim 5, Sharma discloses the parallel processing network of claim 3 wherein the user specified criteria also includes a maximum number of CPUs to be used per machine to execute processes (col. 2 lines 38-62, "The maximum number of execution units is an upper bound to support a peak client load without overloading the server system").

4. Claims 4, 6-9, 12-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Coyle in view of Broadhurst in view of Sharma as applied to claim 3 above, and further in view of Billups, III (USPN 6,173,246) (hereinafter Billups).

As per claim 4, Billups discloses the following limitations not shown by the modified Coyle, specifically the parallel processing network of claim 3 wherein the user specified criteria also includes a model parameter (col. 9 lines 40-55, "The computer resources selected may be the same computer on which the process automation software is executing, or a computer specified on a network by checking a set of criteria such as CPU type").

It would have been obvious to one of ordinary skill in the art to combine the modified Coyle with Billups since Billups provides the added benefit of allowing a CPU type to be specified. This feature allows the parallel processing network to ensure that processes being spawned are spawned in accordance with CPUs that they are capable of executing on. For instance, if a process is spawned that can only be run on a specific type of machine, spawning a process to an incompatible machine would be useless. By indicating the CPU type, a predetermination can be made as to whether or not the process will execute properly.

As per claim 6, Billups discloses the parallel processing network of claim 4 wherein each of the plurality of computers includes a CPU and the model parameter refers to the type of CPU (col. 9 lines 40-55, "The computer resources selected may be the same computer on which the process automation software is executing, or a computer specified on a network by checking a set of criteria such as CPU type").

As per claim 7, Billups discloses the parallel processing network of claim 3 wherein the user specified criteria includes a resource parameter (col. 9 lines 40-55, "The computer resources selected may be the same computer on which the process automation software is executing, or a computer specified on a network by checking a set of criteria such as CPU type, speed, or current load", wherein the criteria specified therein by Billups all refer to different parameters that may be used to describe the system's resources).

As per claim 8, the modified Coyle does not specifically disclose the parallel processing network of claim 7 wherein each of said plurality of computers includes a network interface and the resource parameter refers to a type of network interface.

"Official Notice" is taken that the identification of a network interface in identifying network resources is well known and expected in the art. That is, data communication in a network is encapsulated through layers of protocols, known as the OSI stack. When data travels across a network, the type of network interface must be known to the network in order to properly handle the routing of data. To that end, it would have been obvious to one of ordinary skill in the art to include the network interface type as one of the computer resources identified by Billups since it would allow the network to properly handle all network traffic.

As per claim 9, Billups discloses the parallel processing network of claim 1 wherein said process spawning logic compares the user specified criteria to network features (col. 9 line 40-55, "The computer resources selected may be the same computer on which the process automation software is executing, or a computer specified on a network by checking a set of criteria", wherein the criteria further describes features of the network including specific details concerning each node of the network).

As per claim 12, Billups discloses the parallel processing network of claim 9 wherein each of said plurality of computers includes a CPU and the network features include the model of CPU (col. 9 lines 40-55, "The computer resources selected may be the same computer on which the process automation software is executing, or a computer specified on a network by checking a set of criteria such as CPU type").

As per claim 13, the modified Coyle does not specifically disclose the parallel processing network of claim 9 wherein each of said plurality of computers includes a network interface resource and the network features include the type of network interface resource.

"Official Notice" is taken that the identification of a network interface in identifying network resources is well known and expected in the art. That is, data communication in a network is encapsulated through layers of protocols, known as the OSI stack. When data travels across a network, the type of network interface must be known to the network in order to properly handle the routing of data. To that end, it would have been obvious to one of ordinary skill in the art to include the network interface type as one of the computer resources identified by Billups since it would allow the network to properly handle all network traffic.

As per claim 14, Broadhurst discloses the parallel processing network of claim 9 wherein the user specified criteria includes a number of processes to be spawned (col. 5 lines 43-64, "search engine 226 calculates the number of search sub-processes to spawn based on the number of domains to query submitted by client computer") and, if said spawning logic determines there are insufficient network features to spawn processes in accordance with the user specified criteria, the spawning logic spawns fewer processes than the user specified number of processes (col. 5 lines 43-64, "depending on the load of query server 104, search engine 226 may change the number of search sub-processes to spawn").

5. Claims 10-11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Coyle in view of Broadhurst in view of Sharma in view of Billups as applied to claim 9 above, and further in view of Pearce et al. (USPN 5,910,951) (hereinafter Pearce).

As per claim 10, Pearce discloses the following limitations not shown by the modified Coyle, specifically the parallel processing network of claim 9 wherein the network features are maintained in a process scheduler included in one of said plurality of computers (col. 3 lines 13-

31, "The implementation of this network availability monitor 58 may take several forms so long as the transmit scheduler 62 is provided with sufficient network availability information").

It would have been obvious to one of ordinary skill in the art to add Pearce to the modified Coyle since Pearce provides the additional feature of allowing the process scheduler to have certain information related to the status of the network. Specifically, Pearce refers to an availability monitor within the scheduler that can detect the availability of various nodes on the network. This allows the scheduler to ensure that it does not schedule a process for execution on a node that is currently occupied, thereby preventing stalling of a process due to the unavailability of a resource.

As per claim 11, Pearce discloses the parallel processing network of claim 9 wherein the network features include an identification of which of said plurality of computers is operational and which are nonoperational (col. 4 lines 13-29, "The network availability monitor 58 continually updates the transmit scheduler 62 regarding availability of the qualifying networks", wherein an operational computer could be considered available and a nonoperational computer could be considered unavailable).

Further, Coyle discloses process spawning logic in association with a networking environment. Please see the discussion of claim 1 for citation of how Coyle shows process spawning logic as a network feature.

6. Claims 15 and 20-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Coyle in view of Broadhurst in view of Sharma in view of Pearce.

As per claim 15, Coyle discloses a parallel processing network, comprising:

a plurality of processors coupled together by a communications link (col. 6 lines 52-59, "Distributed system 300 includes three nodes, NODE1, NODE2, and NODE3 that are communicatively coupled so that they can exchange data and information with each other");

Pearce discloses a process scheduler accessible by at least one of said processors, said process scheduler maintains a list of network features (col. 3 lines 13-31, "The implementation of this network availability monitor 58 may take several forms so long as the transmit scheduler 62 is provided with sufficient network availability information");

spawning logic coupled to said process scheduler, said spawning logic receives a set parameters from a user that determine how processes are to be spawned by the root machine, the set of parameters including a user desired number of processes to be spawned ("search engine 226 calculates the number of search sub-processes to spawn based on the number of domains to query submitted by client computer 106"); and

Sharma discloses said spawning logic determines whether a sufficient number of processors are available to permit the user desired number of processes to be spawned in accordance with the user specified parameters, and if not, spawning a reduced number of processes based on the number of available processors (col. 2 lines 38-62, wherein a processor can be interpreted to be one type of execution unit).

It would have been obvious to one of ordinary skill in the art to combine Coyle with Broadhurst since Coyle discloses reducing the number of processes to be spawned in a system, but limits the reducing of processes to situations where system resources are to be saved, rather than utilized to their maximum efficiency. Broadhurst makes up for this by allowing the number of processes to be spawned to be changed depending on the load of the server. The combination of Coyle and Broadhurst would allow a reduction in the number of processes spawned in response to an overload condition. However, this reduction in the number of processes spawned is limited to a single server. Sharma makes up for this deficiency by making a determination if a sufficient number of execution units, e.g. processors, are available to service requests. The combination of Sharma with Coyle and Broadhurst therefore provides a way of ensuring that a sufficient number of processors exist to service processing requests, and if not, reducing the number of processes spawned such that the system is utilized more efficiently. However, the combination of Coyle, Broadhurst, and Sharma does not specifically mention that the monitoring of the network should exist within the confines of a process scheduler. To that end, the addition of Pearce allows the monitoring of network availability within a process scheduler. This provides the added benefit of ensuring that processes are not spawned to nodes that are unavailable for any reason, as well as ensuring that available nodes service requests, thereby reducing the number of idle nodes and making the system more efficient. Therefore, it would have been obvious to combine Coyle, Broadhurst, Sharma, and Pearce for the reasons stated above.

As per claim 20, regarding a computer readable storage for storing an executable set of software instructions which, when inserted into a host computer system, is capable of controlling the operation of the host computer, said software instructions being operable to automatically spawn parallel processes in a parallel processing network, comprising:

Broadhurst discloses a means for receiving user specified criteria (col. 5 lines 43-64, "Upon the user at client computer 106 entering a domain name to search, client computer 106 sends the user input to query server 104 [sep 410]", "search engine 226 calculates the number of search sub-processes to spawn based on the number of domains to query submitted by client computer 106");

Pearce discloses a means for reading a process scheduler to access a list of features associated with the parallel processing network (col. 3 lines 13-31, "The implementation of this network availability monitor 58 may take several forms so long as the transmit scheduler 62 is provided with sufficient network availability information");

the combination of Broadhurst and Pearce discloses a means for comparing the list of network features to the user specified criteria (wherein Broadhurst discloses changing the number of processes to spawn based on the load of a server, and Pearce discloses a network monitor that monitors the availability of network nodes, and the combination thereof would allow for the ensuring of sufficient network availability and changing the number of processes to be spawned based on that availability);

the combination of Sharma and Coyle disclose a means for spawning a number of processes that is reduced based on the number of available CPUs (see discussion of claim 15).

As per claim 21, Broadhurst discloses the computer readable storage medium of claim 20 wherein the user specified criteria includes a user desired number of processes to be spawned ("search engine 226 calculates the number of search sub-processes to spawn based on the number of domains to query submitted by client computer 106") and said means for spawning processes includes a means for spawning the user desired number of processes if said means for comparing determines that the parallel processing network has sufficient features in accordance with the user specified criteria (col. 5 lines 43-64, "depending on the load of query server 104, search engine 226 may change the number of search sub-processes to spawn", wherein if the network features indicate an ability to service the number of processes to spawn, the number of processes spawned is the same as what is specified by the user input).

As per claim 22, Coyle discloses the computer readable storage medium of claim 21 wherein said means for spawning processes includes spawning fewer than the user desired number of processes if said means for comparing determines that the parallel processing network has insufficient features in accordance with the user specified criteria (col. 3 lines 51-67, "Moreover, since in some circumstances, for example when very little work is being performed on a particular node, the number of active processes can be reduced, additional system resources are saved", wherein Broadhurst further discloses reducing the number of processes spawned depending on the load of the server, and the combination thereof meets this limitation).

As per claim 23, Broadhurst discloses the computer readable storage medium of claim 21 wherein said means for spawning processes includes spawning fewer than the user desired number of processes if said means for comparing determines that the parallel processing network has insufficient CPUs to spawn the user desired number of processes (col. 5 lines 43-64, "depending on the load of query server 104, search engine 226 may change the number of search sub-processes to spawn").

7. Claims 16-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Coyle in view of Broadhurst in view of Sharma in view of Pearce as applied to claim 15 above, and further in view of Billups.

As per claim 16, the modified Coyle does not specifically disclose the parallel processing network of claim 15 wherein the user parameters include a particular model of processor to which the processes are to be spawned.

Billups discloses the parallel processing network of claim 15 wherein the user parameters include a particular model of processor to which the processes are to be spawned (col. 9 lines 40-55, "The computer resources selected may be the same computer on which the process automation software is executing, or a computer specified on a network by checking a set of criteria such as CPU type").

It would have been obvious to one of ordinary skill in the art to add Billups to the modified Coyle since Billups provides the added benefit of allowing a CPU type to be specified. This feature allows the parallel processing network to ensure that processes being spawned are spawned in accordance with CPUs that they are capable of executing on. For instance, if a process is spawned that can only be run on a specific type of machine, spawning a process to an incompatible machine would be useless. By indicating the CPU type, a predetermination can be made as to whether or not the process will execute properly.

As per claim 17, the modified Coyle does not specifically disclose the parallel processing network of claim 16 wherein the user parameters include a particular type of a network resource.

"Official Notice" is taken that the identification of a particular network resource is well known and expected in the art. That is, data communication in a network is encapsulated through layers of protocols, known as the OSI stack. When data travels across a network, the type of network resources must be known to the network in order to properly handle the routing of data. To that end, it would have been obvious to one of ordinary skill in the art to include the network resource type as one of the computer resources identified by Billups since it would allow the network to properly handle all network traffic.

As per claim 18, Pearce discloses the parallel processing network of claim 17 wherein the spawning logic determines whether sufficient network features are available to permit the user desired number of processes to be spawned by accessing the process scheduler to read the list of network features (col. 3 lines 13-31, "The implementation of this network availability monitor 58 may take several forms so long as the transmit scheduler 62 is provided with sufficient network availability information", wherein the scheduler is continually updated as to the availability of network stations, and this can be relayed to any node of the network that desires to know the availability of the different networks or nodes).

As per claim 19, Sharma discloses the parallel processing network of claim 17 wherein the user parameters include a maximum number of CPUs to use per machine for spawning processes (col. 2 lines 38-62, "The maximum number of execution units is an upper bound to support a peak client load without overloading the server system").

8. Claim 24 is rejected under 35 U.S.C. 103(a) as being unpatentable over Broadhurst in view of Pearce in view of Sharma.

As per claim 24, regarding a method of creating processes in a multi-processor network, comprising:

Broadhurst discloses receiving criteria that determine how the processes are to be created, the criteria including a desired number of processes to be created (col. 5 lines 43-64, "Upon the user at client computer 106 entering a domain name to search, client computer 106 sends the user input to query server 104 [sep 410]", "search engine 226 calculates the number of search sub-processes to spawn based on the number of domains to query submitted by client computer 106");

the combination of Pearce and Sharma discloses comparing the criteria to a database of network features (Pearce, col. 4 lines 13-29, "When the network availability monitor 58 detects a newly available network, the transmit scheduler 62 scans the transmit database for transmission requests") to determine if there are a sufficient number of processors to accommodate the desired number of processes (Sharma, col. 2 lines 38-62, wherein a processor can be interpreted to be one type of execution unit); and

Broadhurst discloses creating processes if a sufficient number of processors are available and, if a sufficient number of processors are not available, creating a reduced number of processes (col. 5 lines 43-64, "depending on the load of query server 104, search engine 226 may change the number of search sub-processes to spawn", wherein the determination of how many processes to spawn is determined by the availability of network resources).

The motivation for combining Broadhurst, Sharma, and Pearce can be considered to be essentially the same as the motivation for combining Coyle, Broadhurst, Sharma, and Pearce, as discussed in reference to claim 15. Coyle merely presents the added feature of reducing the number of processes to suit the system's needs and can thus be considered an expansion upon the idea presented by Broadhurst.

9. Claims 27-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Broadhurst in view of Sharma in view of Pearce as applied to claim 25 above, and further in view of Billups.

As per claim 27, Billups discloses the following limitations not shown by the modified Broadhurst, specifically the method of claim 24 wherein receiving criteria includes receiving criteria that also include a model of processor and a resource type for running processes (col. 9 lines 40-55, "The computer resources selected may be the same computer on which the process automation software is executing, or a computer specified on a network by checking a set of criteria such as CPU type").

It would have been obvious to one of ordinary skill in the art to combine the modified Broadhurst with Billups since Billups provides the added benefit of allowing a CPU type to be specified. This feature allows the parallel processing network to ensure that processes being spawned are spawned in accordance with CPUs that they are capable of executing on. For instance, if a process is spawned that can only be run on a specific type of machine, spawning a process to an incompatible machine would be useless. By indicating the CPU type, a predetermination can be made as to whether or not the process will execute properly.

As per claim 28, the modified Broadhurst does not specifically disclose the method of claim 27 wherein the resource type includes a network interface resource type.

"Official Notice" is taken that the identification of a network interface in identifying network resources is well known and expected in the art. That is, data communication in a network is encapsulated through layers of protocols, known as the OSI stack. When data travels across a network, the type of network interface must be known to the network in order to properly handle the routing of data. To that end, it would have been obvious to one of ordinary skill in the art to include the network interface type as one of the computer resources identified by Billups since it would allow the network to properly handle all network traffic.

10. Claims 29-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Broadhurst in view of Cohoe et al. (USPN 6,108,309) (hereinafter Cohoe).

As per claim 29, Broadhurst discloses a method for spawning processes in a multiprocessor network, comprising:

specifying whether processes are to be spawned automatically to match a set of criteria (col. 5 lines 43-64, "Upon the user at client computer 106 entering a domain name to search, client computer 106 sends the user input to query server 104 [sep 410]", "search engine 226 calculates the number of search sub-processes to spawn based on the number of domains to query submitted by client computer 106", wherein the spawning of processes is automatically adjusted to account for the load on the server);

spawning processes to match the criteria if automatic spawning is specified (see above citation); and

spawning a reduced number of processes automatically based on a number of available processing resources (col. 5 lines 43-64, "depending on the load of query server 104, search engine 226 may change the number of search sub-processes to spawn", wherein the determination of how many processes to spawn is determined by the availability of network resources)

Cohoe discloses the following limitations not shown by Broadhurst, specifically spawning processes in accordance with a process group file or spawning processes in accordance with the process group file if so specified (col. 11 lines 38-58, "I/O process spawns the CRP and AMP process threads which continuously run in parallel", "the CRP process reads its configuration files and, in response to the initialization data, the CRP attaches to the shared memory allocated...", wherein the process is spawned based on a configuration file and that file determines how memory is allocated and what data structures the process is to use).

It would have been obvious to one of ordinary skill in the art to combine Broadhurst with Cohoe since Cohoe provides an additional way of automating the process spawning. That is, rather than requiring user input to determine how processes are to be spawned, a configuration file could be used to tell the system how to spawn processes. In that sense, the group file could be overridden by the user-defined criteria. This adds additional functionality and flexibility to the system. The combination of Broadhurst and Cohoe would permit the automatic spawning of a reduced number of processes based on the information in a process group file, wherein Broadhurst supports spawning a reduced number of processes based on the current load of those processing resources.

As per claim 30, Broadhurst discloses the method of claim 29 further including determining whether the multiprocessor network matches the set of criteria if automatic spawning is specified (col. 5 lines 43-64, "search engine 226 calculates the number of search sub-processes to spawn based on the number of domains to query submitted by client computer", wherein the multiprocessor network matches the criteria if sufficient processing resources are available on the network, and if not, the number of processes spawned is changed to meet those resource requirements).

Response to Arguments

11. Applicant's arguments filed September 23, 2003 have been fully considered but they are not persuasive.

12. Applicant argues on page 9 that the combination of Coyle, Broadhurst, and Sharma fails to meet the requirements of claim 1, since "*the load of query server 104* [in Broadhurst] *does not represent or even suggest the number of available processors, as required by claim 1. As one of ordinary skill in the art is aware, the load of a server is different than the number of available processors*" or that "*Broadhurst does not teach or even imply the use of more than one*

processor." Applicant presents similar arguments on pages 10-12 in reference to claims 15, 20, 24, and 29.

Regarding the latter part of the argument, that Broadhurst does not teach or even imply the use of more than one processor, attention is directed to the following quotation within Broadhurst, which leaves no doubt that multiple processors are supported (col. 7 lines 9-26, "Methods consistent with the present invention also work well with multi-processor machines. On a multi-processor machine, a search may show up one more than one processor thread").

Regarding the first part of the argument, that the load of the query server in Broadhurst does not represent or suggest the number of available processors, Examiner respectfully disagrees. Specifically, when the combination of references is considered, as opposed to Broadhurst alone, the spawning of a reduced number of processes based on the number of available processors would have been obvious to one of ordinary skill in the art. That is, Broadhurst refers to querying a server to determine its processing load, and changing the number of processes to spawn based on that load. Also, as noted above, Broadhurst provides for a multi-processor machine. Thus, the querying of a multi-processor machine would allow a determination to be made as to the available processing load of each processor, particularly when considered in combination with Sharma, which discloses determining a processing load of a particular execution unit, and rejecting the process if it cannot service the request adequately.

13. Applicant argues on pages 9-10, "*Sharma does not teach, suggest, or even imply that the user parameters include the maximum number of CPUs, as required by claim 5.*" Applicant presents a similar argument on pages 10-11 in reference to claim 19.

This argument hinges on the fact that an execution unit as defined by Sharma does not suggest a CPU. In particular, Sharma discloses including a maximum number of **execution units**, wherein the maximum number of execution units is determined such that overload conditions are prevented. This is similar to what is claimed in the sense that spawning a reduced number of processes is intended to reduce the overall computational load such that an overload is prevented. The term "execution unit" could refer to a processor, thread, or stand-alone machine, i.e., any "execution unit" capable of servicing a request. Thus, Sharma meets the limitation argued in claim 5.

Conclusion

14. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J Ali whose telephone number is (703) 305-8106. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William A Grant can be reached on (703) 308-1108. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Syed Ali
December 11, 2003

WILLIAM GRANT
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100
12/12/03